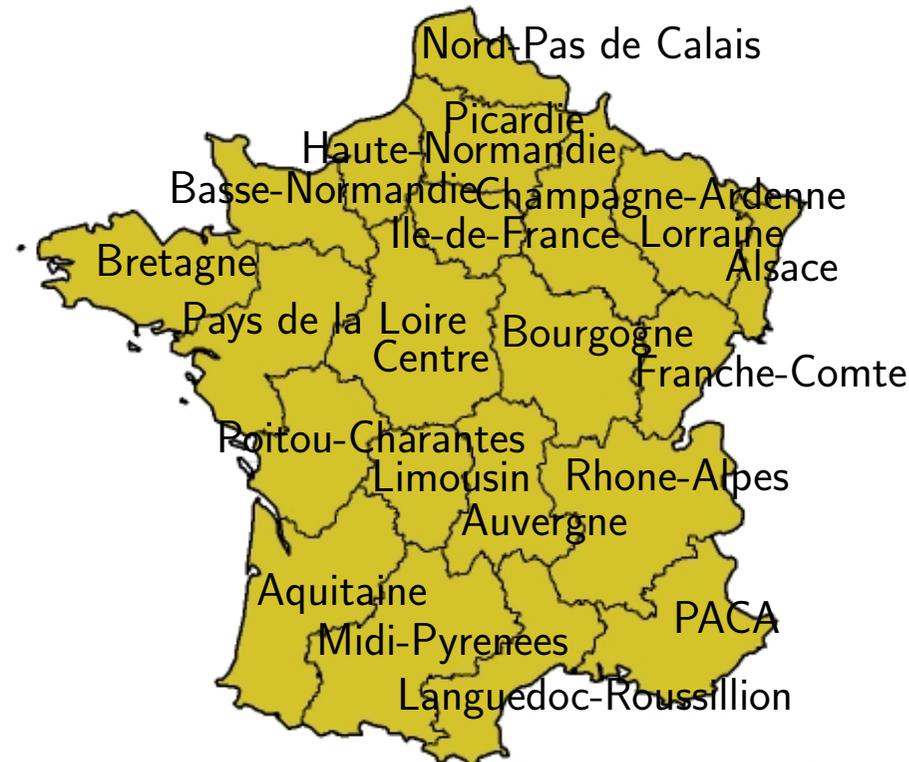


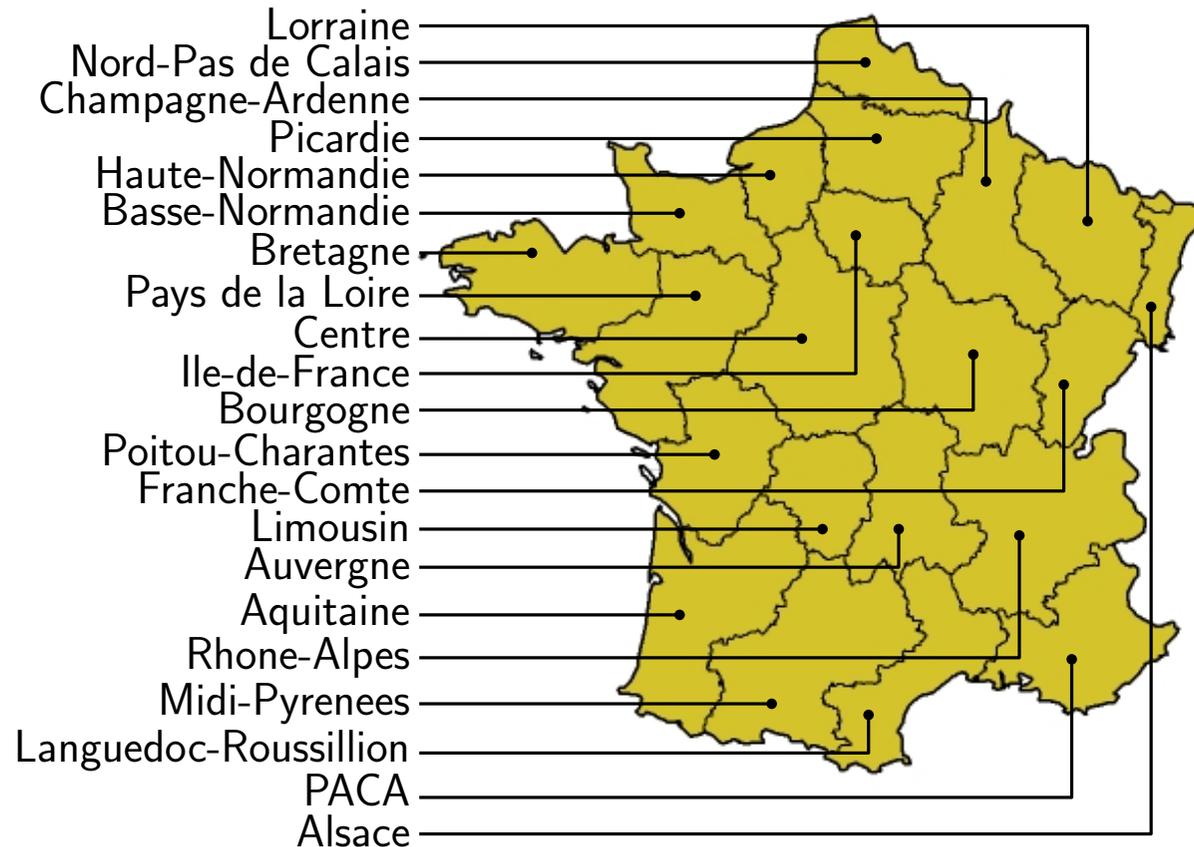
Map Labeling, U-Bahnpläne und dynamische Landkarten

Alternatives Modell: Boundary Labeling



- „Traditionelles“ Map Labeling führt nicht immer zu besten Resultaten
- dichte Regionen oft nicht vollständig beschriftbar

Alternatives Modell: Boundary Labeling



- „Traditionelles“ Map Labeling führt nicht immer zu besten Resultaten
- dichte Regionen oft nicht vollständig beschriftbar
- Idee: Platziere Label am Rand und verbinde diese mit den Punkten

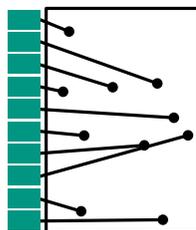
Boundary Labeling (allgemeine Formulierung)

Geg: Punkte $\{p_1, \dots, p_n\}$ in einem Rechteck $\mathcal{R} \subset \mathbb{R}^2$ und Beschriftungsrechtecke $\{R_1, \dots, R_n\}$

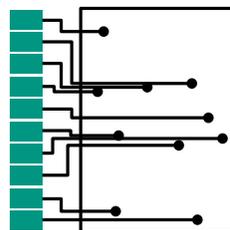
Ges: Platzierung der Rechtecke am Rand von \mathcal{R} ;
für alle i eine Kurve (*leader*) l_i , die p_i mit R_i verbindet

Weitere mögliche Voraussetzungen:

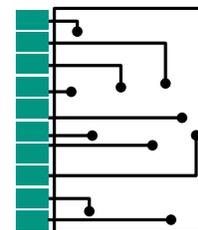
- Beschränkung auf 1/2/4 Seiten von \mathcal{R}
- gleich große Rechtecke R_1, \dots, R_n
- Einschränkung des leader-Typs, z.B. 1 Knick und achsenparallel
- keine Kreuzungen der leader
- Minimierung einer geeigneten Kostenfunktion, z.B. der Gesamtlänge der leader oder der Anzahl Knicke



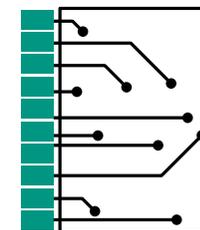
straight



Typ *opo*



Typ *po*

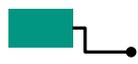
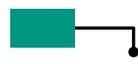


Typ *do*

Boundary Labeling: was weiß man?

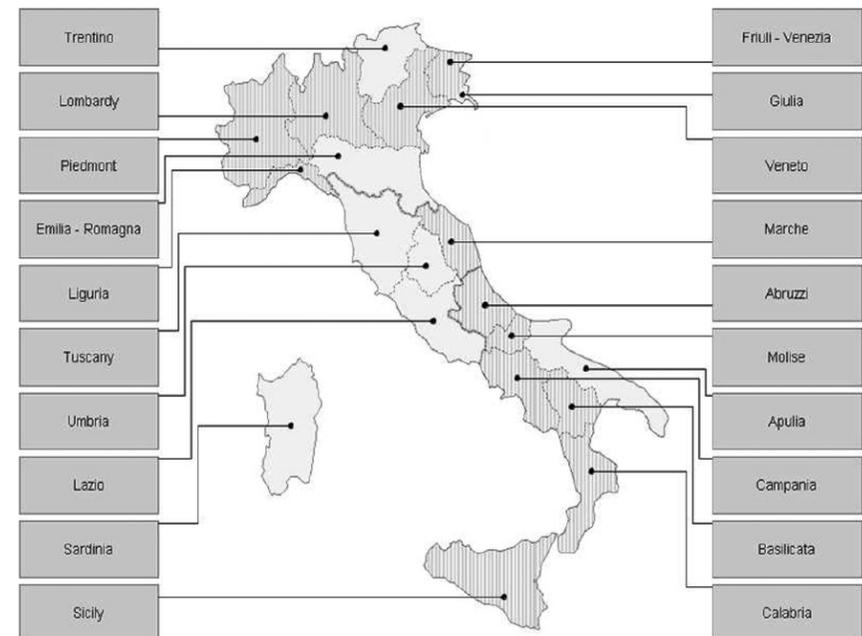
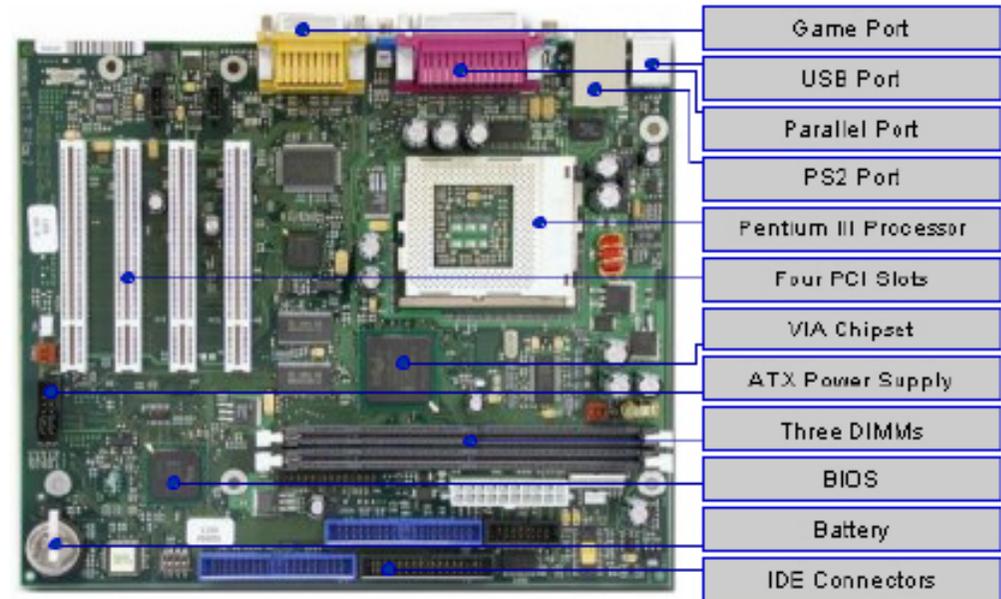
Minimierung der Gesamtlänge

leader-Typ

	straight 	<i>opo</i> 	<i>po</i> 	<i>do/pd</i> 
1	$O(n^{2+\varepsilon})$	$O(n \log n)$	$O(n \log n)$	<i>do</i> : $O(n^2)$ *
2		$O(n^2)$ *	$O(n^2)$	<i>do/pd</i> : $O(n^3)$
4	$O(n^{2+\varepsilon})$	$O(n^2 \log^3 n)$		<i>od/pd</i> : $O(n^3)$

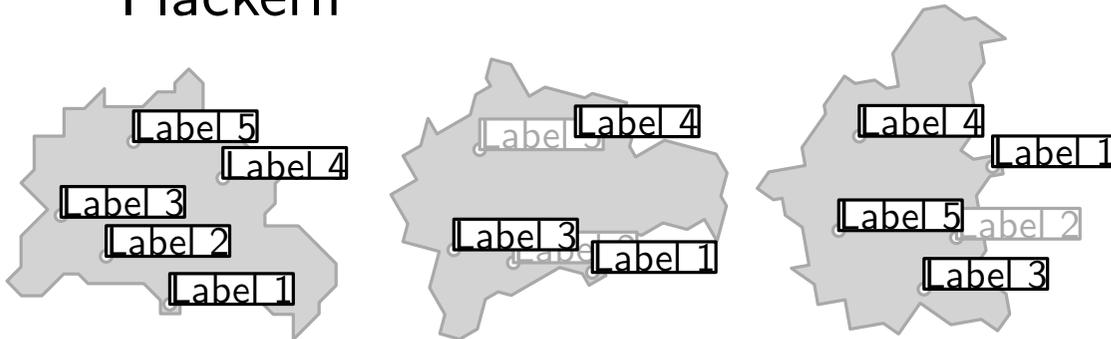
* NP-schwer für ungleiche Labelrechtecke

Beispiele für Boundary Labeling



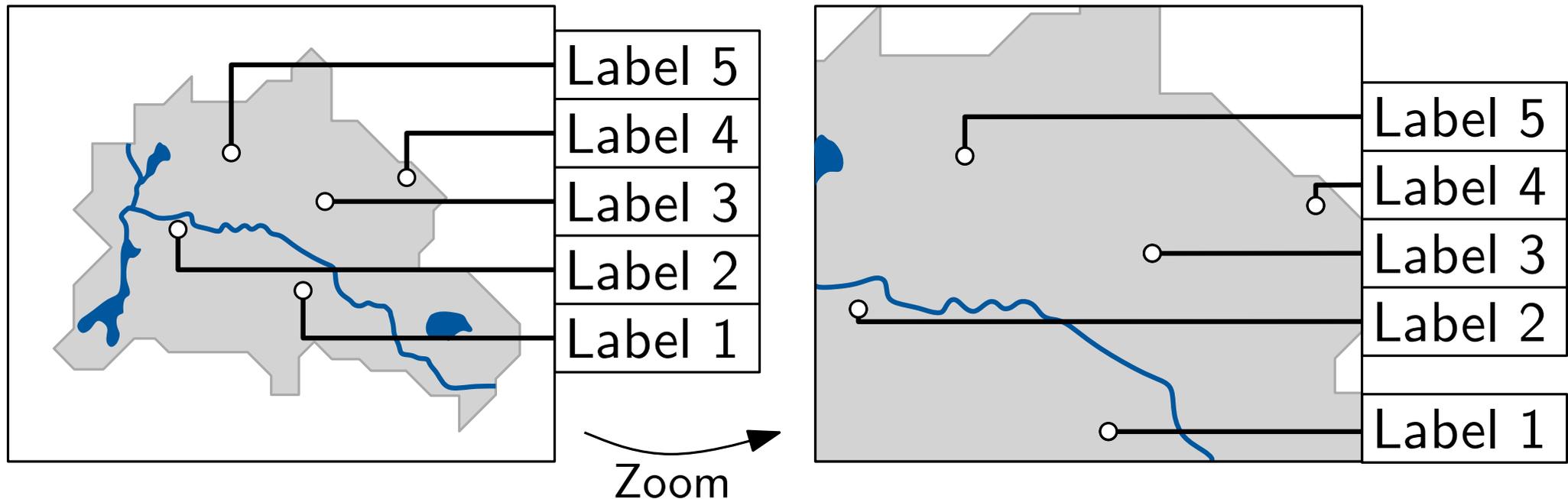
Beschriftung dynamischer Landkarten

- dynamische Landkarten erlauben kontinuierliches Zoomen, Verschieben und Rotieren
- Beschriftung soll *konsistent* bleiben, d.h. kein Springen und Flackern



- Maximierung der Anzahl sichtbarer konsistenter Label ist NP-schwer, auch schon für reines Zoomen bzw. Rotieren
- es gibt jedoch Approximationsalgorithmen für mehrere Varianten

Dynamisches Boundary Labeling

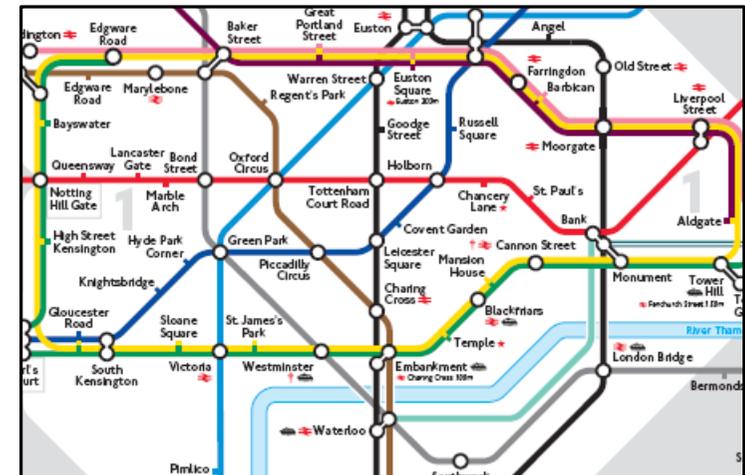
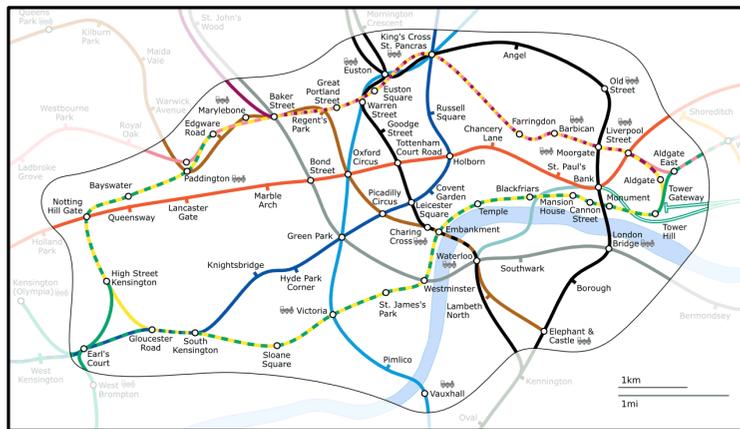


- Ziel: längenminimale Beschriftung für jede mögliche Zoomstufe
- optimale Position der Label ändert sich beim Zoomen
- Berechnung der optimalen Position durch geometrische Verfahren

interaktives Java-Applet unter
<http://www.cs.helsinki.fi/group/compgeom/boundarylabeling>

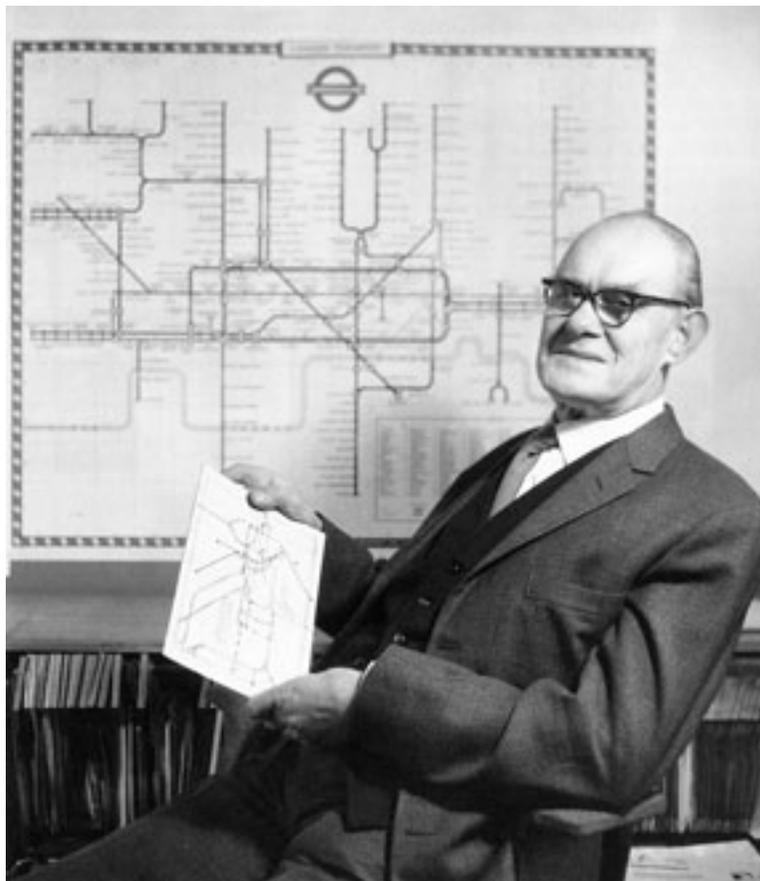
U-Bahnpläne

- Ziel: Navigation für Fahrgäste erleichtern
- visuelle Routenplanung
 - "Wie komme ich von A nach B?"
 - "Wo muss ich aus- oder umsteigen?"
- Geometrie und Maßstab dürfen zur Verbesserung der Lesbarkeit verzerrt werden

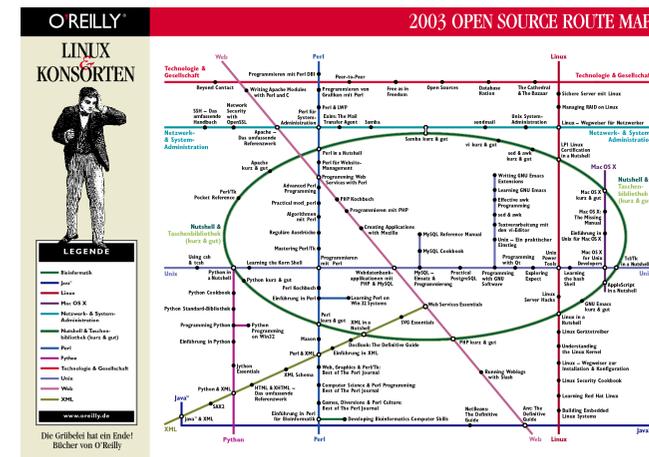
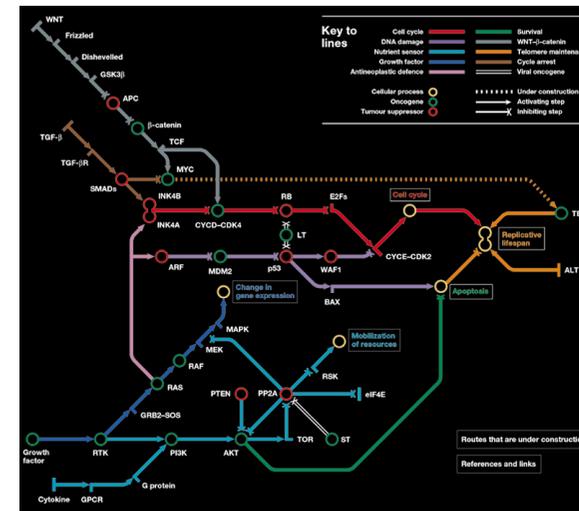


Wozu automatisches Zeichnen?

- Pläne werden bis heute manuell entworfen
- Werkzeug zur Unterstützung der Designer
- Metapher zur Visualisierung abstrakter Daten



Henry Beck, Vater der Londoner Tube Map



Metro Map Layout

Geg: Zeichnung eines Graphen $G = (V, E)$ mit Haltestellen V und Menge \mathcal{L} von Verkehrslinien (Pfaden)

Ges: schematische Zeichnung von G , die

- harte Anforderungen garantiert und
- weiche Anforderungen optimiert

Harte Anforderungen:

- Erhalt der Topologie der Zeichnung
- **oktilineare** Kanten
- Mindestabstände zwischen Stationen
- überlappungsfreie Beschriftung



Metro Map Layout

Geg: Zeichnung eines Graphen $G = (V, E)$ mit Haltestellen V und Menge \mathcal{L} von Verkehrslinien (Pfaden)

Ges: schematische Zeichnung von G , die

- harte Anforderungen garantiert und
- weiche Anforderungen optimiert

Weiche Anforderungen:

- Minimierung von Linienknicken
- Minimierung des Flächenbedarfs
- Erhalt relativer Positionen



Metro Map Layout

Geg: Zeichnung eines Graphen $G = (V, E)$ mit Haltestellen V und Menge \mathcal{L} von Verkehrslinien (Pfaden)

Ges: schematische Zeichnung von G , die

- harte Anforderungen garantiert und
- weiche Anforderungen optimiert

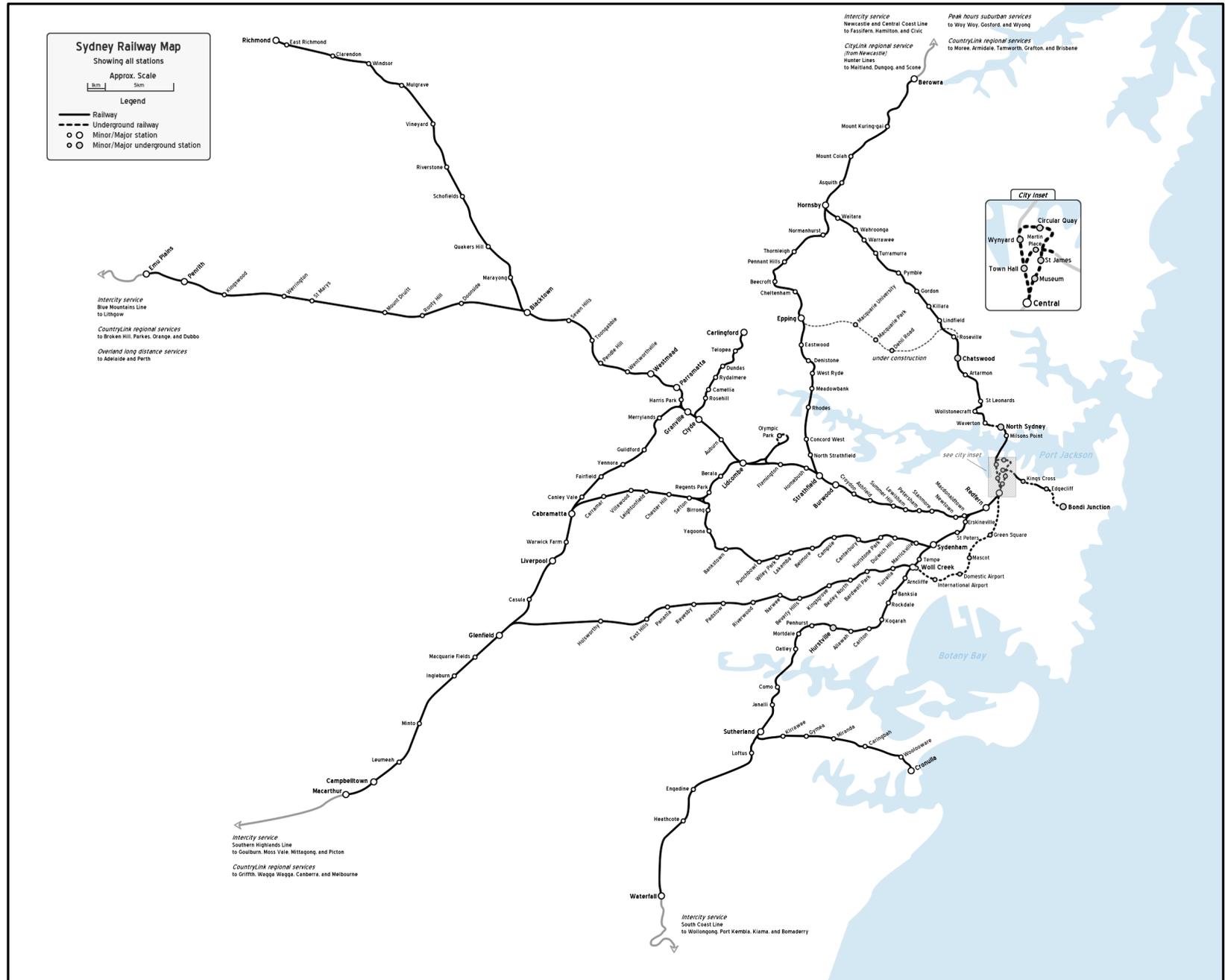
Satz

Das Metro-Map-Layout Problem ist NP-vollständig.

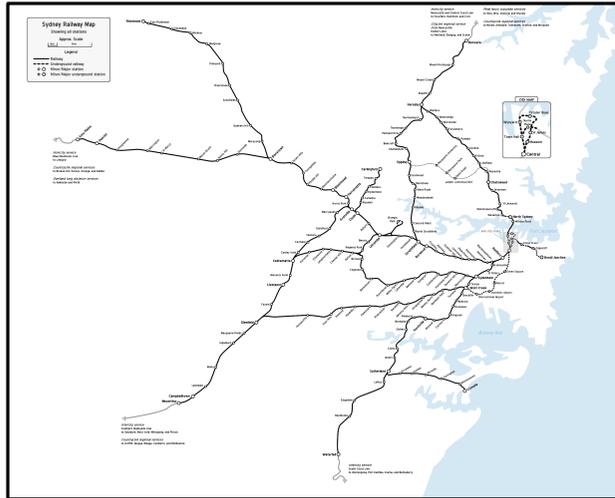
Es lässt sich allerdings als sogenanntes gemischt-ganzzahliges lineares Programm formulieren und lösen.

Beispiel Sydney

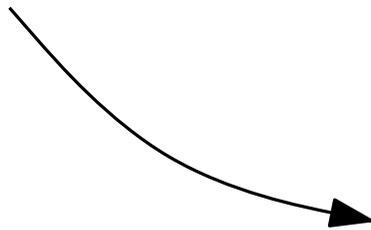
Eingabe:



Beispiel Sydney



Eingabe



Ausgabe nach ca.
12 Stunden

